Automata at Court: Characters, Intelligences, Machines

Adil Bokhari ► P. 210 *Agostino Nickl* ► P. 213

Endnote 1

Introduction

The rapid advancement of artificial intelligence and its many faces and the recent mass adoption in the form of Large Language Models (LLMs) mark another shift in our way of seeing and talking to the world. Already today, we see design disciplines as well as the role of the architect challenged by machines that can simulate and modulate existing mechanics and media. But transformative changes are on the horizon well beyond the field of architecture and design: LLMs consolidate both specialist and general knowledge on a new scale – a moment in which some experts see the genesis of Artificial General Intelligence (AGI). While the debate rages on whether such future AGI and the sweeping automation it affords will be a blessing or curse, and AI's future impact on the role of designers and architects is still unclear, this essay aims to offer temporary respite, enacting a coded thought experiment with and by AI. *Automata at Court* invites us to move away from thinking of chatbot interfaces as the singular mode of human–machine interactions, proposing *prompting cascades* as an element of architectonic abstraction through code. It playfully questions to what extent the roles of architects and designers can be role-played by LLMs. Finally, it explores what happens when hundreds of automata role-played by AI are let loose on the Palace of Versailles to stage discussions on architecture, intelligence, and technology. While applied to a fictional setting, Automata at Court draws out pertinent questions on how we, as architects and designers, can come to terms with new forms of such general intelligence.

Contracting Machine Intelligence(s):

The idea here is not to collapse machine intelligence and human intellect into one or to build hierarchies between them, but to establish a *logistic* framework – related to 'not primarily logics, but logistics' (Hovestadt 2013, 65). Here, many of these distinctions and roles might be hosted, and contracts between different intelligences can be *continuously negotiated*.

The LLM as Foundation Model

LLMs like OpenAI's ChatGPT (OpenAI 2022) – showcase outstanding capabilities across numerous language-oriented tasks, such as understanding instructions and generating responses in natural language. Utilising transformer architectures and efficient stacks of neural networks making use of self-attention, these models can capture long-range dependencies in sequential data and coherently continue input sequences (Vaswani et al. 2017). Trained on vast quantities of textual data as well as on interactions with human users (Reinforcement Learning from Human Feedback – RLHF) allows them not only to internalise complex patterns, syntactic structures and contextual nuances, but also to effectively follow prompts, displaying emergent learning and reasoning capabilities (Bubeck et al. 2023). When combined with neural networks for image generation, LLMs excel in text-to-image generation tasks (Croitoru et al. 2023; Kang et al. 2023). Noteworthy examples include DALL-E, Stable Diffusion, and Midjourney (OpenAI 2021; Stability AI 2023; Midjourney Inc. 2023), which have already found applications in design practice (Galanos, Liapis, and Yannakakis 2023; Köhler 2022; Leach 2022). But beyond text and image generation, LLMs are also able to undertake a wide range of tasks online – from teaching online classes to performing complex web-based tasks autonomously, as demonstrated by AutoGPT (Significant-Gravitas 2023; AutoGPT 2023). The general intelligence of LLMs makes them ideal foundation models for building a myriad of different autonomous and automated applications. This new scale of automation rekindles anxieties related to faceless machines that we do not fully comprehend (Srivastava et al. 2022).

Automation and Anxiety

The idea of machinic automation and our relationship to it has been troubled, to say the least – from the invention of the robot as 'slave' (R.U.R.) (Endnote 2) to eschatological prophecies of a technological singularity (Capek 2004; Harper 2023b; Kurzweil 2005; Vinge 1993; Hui 2023). Paradoxically, we tend to ascribe to machines the role of a "sacred object", an ascription that 'gives rise to an intemperate technicism'; but, haunted by the 'permanent danger of aggression and insurrection', we also banish them to eternal servitude, *"placing machines in the service of man"* (Simondon 2017, 16-17).
Such short-sighted discussions on automation *"fall into a determinist trap of thinking that [these technologies] wrestle control over life from humans. Instead, they operate in relations with bodies and environments, with questions of control and agency failing to capture the relational and distributed ways they operate"* (Tucker 2021). While 'pre-cybernetic machines could be haunted' but were, strictly speaking, *"not self-moving, self-designing, autonomous",* with today's machines *"we are not so sure ... Our machines are disturbingly lively, and we ourselves frighteningly inert"* (Haraway 1991, 152). Beyond a paralysis or a fetishisation of their errors and glitches, an abstraction is required to communicate with these new machines with their whims, fancies, idiosyncrasies and 'unique characteristics' (Bang et al. 2023, 3). Here, the age-old notion of the automaton can provide insight.

The Automaton: Mythical, Mechanical, Mathematical

Automata as self-thinking or self-acting beings have intrigued human imagination across various cultures for millennia (Harper 2023). They feature prominently in mythological narratives – including Homer's accounts of *animate statues* from Hephaistos' workshop, as well as in mechanical thought models and mathematical treatises (Cholodenko 2007; Nelson 2016). Their significance spans historical periods – from Classical Antiquity and the Islamic Golden Age to the Renaissance, from Baroque courts to the academies of the early Enlightenment (Harenberg 2015; Schaffer 1999). Automota have characteristically appeared within multiple cultures of the world, where they embody cosmologies and symbolise contracts with the state-of-the-art technologies of the time. Giving a figuration to functional roles such as mechanical timekeepers, they render characteristic behaviours, placing their respective functionalities in a larger philosophical, cultural and historical setting. One of the most famous automata, the Strasbourg Clock, is *"much more than a chronometer; instead, it theatrically stages the cosmic order at work in the mechanical philosophy through the dramaturgy of technological spectacle"* (Reilly 2011, 51). By addressing abstract questions throughout history, from Al-Jazari's Elephant Clock [Fig. 32, left] to John Conway's Game of Life, automata do more than just link technological innovations to the past (Izhikevich, Conway, and Seth 2015). They bring ancient concepts to bear on contemporary discussions, enriching our understanding of both; addressing scientific, technological, mathematical and cultural *motifs* at the same time.

Fig. 32 P. 90–91

Automata as Characterizations of Technologies

We propose to think of automata less as an anthropomorphic *personification of automations* (e.g. robots), and more as embodied *characterisations* of *constellations of technologies*. Through automata, technological innovations can be brought to a table around which they can be discussed, and abstract philosophical questions posited. A mechanically defecating duck [Fig. 32, center-top] asks what it means to be alive; a mechanically drawing draughtsman [Fig. 32, center-bottom] asks what it means to memorise; and a mechanically moving Karakuri doll [Fig. 32, right] asks what it means to 'go as human' (Riskin 2003; The Franklin Institute 2023; Guth 2019; Jullien 2015, 7). A figure, a role and a location are introduced to the technologies, transcending their means of construction and purely pragmatic concerns. They become characters on stage, where they can play in relation to others – the Elephant Clock was placed within the Artuklu Palace, enacting cultures of the known world in a single body with intricate mechanics (Robinson 2007). Beyond the totalising notion of automation, automata work with paradoxes: both closed and open, animate and inanimate, stationary and mobile (Simondon 2017; Cholodenko 2007). In accepting their ambiguity, we find in them individuals who invite us to position ourselves in relation to them. Can ancient automata help us in thinking about the LLMs of today differently?

Thinking LLMs as Automata

An LLM can be seen as a complete individual – as running a machine learning model only partially will not work – and they are often addressed as single, specific *characters*: as chat-bots, co-pilots and side-kicks (OpenAI 2022; GitHub 2023; Snapchat 2023). On the other hand, LLMs, with their capability of "task-agnostic, few-shot performance", can also individuate into multiple roles when fine-tuned on specific datasets (Brown et al. 2020). While BloombergGPT is a model trained on financial news and jargon and sentiment analysis, Marv the Sarcastic Chatbot mimics natural language mannerisms (Wu et al. 2023; Dilmegani 2023; OpenAI 2023). This demonstrates multiple scales of individuation afforded by LLMs in particular – as an automaton itself, it can play several automata. Embedding such a role-play into a larger play can be used to abstract from the logics of pure automation and current, chatbot-based use paradigms to envision instead a novel logistic framework, where constellations of multiple models and levels of abstraction are set in motion as characters.

Automata at Court: An Architectonic Stage Play

Automata at Court is an experiment in deploying LLMs as role-playing automata to re-inhabit a historic architecture, and an architectonic "exercise of building active state's" (Bühlmann 2020, 170). It comprises the following:

- 32 Roles: 32 notable architects as well as 20 expert domains involved in inhabiting spaces
- 32 Places: Individual chambers of the Palace of Versailles as speculative sites
- 32 Motifs reflecting historic motifs addressed by automata
- 1 Table serving as a receptacle for generated contributions, stored as a digital spreadsheet
- Wolfram Mathematica as a coding environment
- GPT 3.5 turbo accessed over OpenAI API for text generation, Midjourney V5 for image generation

Coding Tables

Used for logic and multiplication, for recording the Babylonian night sky, for orchestrating pyramid construction or for plotting novels, our experiment also instrumentalizes the *table* for its logistic framework [Fig. 33] (Huber et al. 1982; Tallet 2017; Joyce 1930). Its rows stand for 32 chosen locations in the Palace of Versailles, to which we allocated the name of a notable architect and a leading motif taken from historic automata. The columns stand for multiple roles and trades needed to inhabit the spaces, spanning expert domains involved in architectural design. The cells thus formed are algorithmically filled by the LLM, following a cascade of prompts. Here, the table fully embraces the ambiguity of being both a 'multiplication table and the dinner table', staging a new kind of role-play – "around the table, only sage tongues" (Deleuze 1990, 64; Serres 2016, 164).

Fig. 33 P. 96–97

Playing Roles

Fig. 35 P. 60

To individuate the LLM into characteristic automata, we focused on the *one* and *few-shot prompting* role-play method, combining a specific *role* (e.g., architect) with a specific figure (e.g., Le Corbusier); only the *figure* of the architect, the *motif* to be addressed, and the *location* of their intervention at Versailles are pre-determined [Fig. 35]: this *automaton* then suggests a theme and title for an event, as well as an *appointee* with whom they propose a program. To complement these two more general characters, we have set 20 more specialised roles from which the architect and the appointee can choose and suggest figures appropriate to the location, theme, title and program of the event [Fig. 35]. Again role-played by the LLM, these characters suggest in parallel and independently from each other individual contributions covering different facets of the space – including visual (e.g., artist), gustatory (e.g., chef), olfactory (e.g., perfumer), auditory (e.g., musician), and tactile (e.g., interior designer) faculties. Their contributions are featured on numbered invitation cards, several of which appear in the figures illustrating this chapter.

Placing Games

The Palace of Versailles serves as the setting for our thought experiment – a place of absolute power and a kaleidoscope of knowledge, of exuberance and restraint, both troubling and intriguing, a world of its own and a model of the world. As a model for roles, mannerisms, rituals and spatialities, the Palace of Versailles sparked architectural imagination – from the social utopias of Charles Fourier's Phalanstères to Corbusier's Unité d'Habitation, from Lucio Costa's modernist urban planning for Brasilia to the postmodern expressionism of Ricardo Bofill for Les Espaces d'Abraxas (Höpel 2012; Blair 2012; Farhat 2023; Picon 2017). Where once mechanical automata came to flourish in the theatre of courtly life, now 32 characteristic indoor and outdoor spaces from its current layout are brought back to life by digital automata showcasing their *"capacity for animation that passes from human beings to things and from things to human beings"* (Coccia 2020). Often mythologically charged, the names of the chosen locations in Versailles are placed in conversation with the role-playing LLM.

Prompting Cascades

To programmatically inhabit the Palace, we chose the highly abstract Wolfram Language to code tabular *prompting cascades*. At their core is a simple function, taking a *text-prompt* as a string and a *temperature* hyperparameter controlling the model's determinacy as arguments and passing them on to the GPT 3.5 turbo model via the OpenAI API (OpenAI 2023). This function is then called upon in fixed prompting templates with placeholders for pre-determined and generated values [Fig. 35]. Prompts start relatively simple and become gradually more complex as they incorporate previously given answers. Key responses are then stored in a table represented as a data structure in Mathematica. The main technical challenge for coding the play lies in the dif-

PREREQUISITES

01	Billiard Room	Claude Nicolas Ledoux	Force
02	Bosquets Forests	Julia Morgan	Cosmos
03	Cabinet of Agates	Norma Merrick Sklarek	Intimacy
04	Cabinet of Jewels	Margarete Schütte-Lihotzky	Information
05	Cabinet of Medals	Carlo Scarpa	Storytelling
06	Geography Cabinet	Aldo Rossi	Cartography
07	Labyrinth	Luis Barragán	Memory
08	Lantona Fountain	Francesco Borromini	Transformation
09	Marble Court	Marcus Vitruvius Pollio	Geometry
10	Garden Ballroom	Zaha Hadid	Ceremony
11	Hall of Mirrors	Robert Venturi	Magic
12	King's Bedchamber	Cini Boeri	Time
13	King's Vegetable Garden	Frank Lloyd Wright	Agriculture
14	Marble Staircase	Antoni Gaudí	Gravity
15	Neptune Fountain	Lina Bo Bardi	Continuity
16	Orangery	Michael Graves	Logistics
17	Peace Salon	Paul Revere Williams	Communication
18	Queen's Bedchamber	Andrea Palladio	Ritual
19	Queen's Guard Room	Alison Smithson	Order
20	Royal Chapel	Le Corbusier	Virtue
21	Royal Opera	Eileen Gray	Imagination
22	Salon of Abundance	Filippo Brunelleschi	Life
23	Salon of Apollo	Buckminster Fuller	Energy
24	Salon of Diana	Gian Lorenzo Bernini	Nature
25	Salon of Hercules	Donato Bramante	Power
26	Salon of Mars	Aldo van Eyck	Game
27	Salon of Mercury	Leon Battista Alberti	Cryptography
28	Salon of Venus	Mies van der Rohe	Engendering
29	Small Gallery	Ray Eames	Myth
30	Tribune of Musicians	Adolfo Natalini	Synchronicity
31	War Salon	Oscar Niemeyer	Diplomacy
32	Vestibule to the Chapel	Kenzo Tange	Purification

PROMPT CASCADE TEXT

```
1     Initialize rooms as a list of location names or descriptions
2     Initialize architects as a list of architect names as figures
3     Initialize experts as a dictionary with roles as keys and attributes as values
4     Initialize motifs as a list of motif descriptions
5     Initialize temp as a value between 0.0 and 2.0

6     For each i in number of rooms:
6.1       room[i].theme =          AskGPT("What theme would architects[i] propose to inhabit rooms[i],
                                   addressing the motif of motifs[i]?", temp)
6.2       room[i].event =          AskGPT("What event would architects[i] choose to bring rooms[i] to
                                   life under the theme of room[i].theme addressing the motif of
                                   motifs[i]?", temp)
6.3       room[i].appointee =      AskGPT("Who would architects[i] choose to run the event
                                   room[i].event in rooms[i] under the theme of room[i].theme while
                                   addressing the motif motifs[i]?", temp)
6.4       room[i].program =        AskGPT("What program might the event room[i].event with the theme
                                   room[i].theme organized by architects[i] and room[i].appointee
                                   in rooms[i] entail?", temp)
6.5       room[i].roles =          AskGPT("What roles would architects[i] and room[i].appointee choose
                                   to host room[i].event in rooms[i] considering the program
                                   room[i].program? Pick at least 8 roles.", temp)
6.6       room[i].figures =        AskGPT("Who might room[i].appointee choose to fill the roles for
                                   room[i].roles, formatted as 'role : name'?", temp)
6.7       room[i].suggestions =    Split room[i].figures by newline

6.8       For each suggestion in room[i].suggestions:
6.8.1         characters = Split suggestion by ":"
6.8.2         If characters[0] is part of experts:
6.8.3             room[i].experts[characters[0]] = characters[1]

6.9       For each j in number of room[i].experts:
6.9.1         room[i].experts[j].contribution =
                                   AskGPT("What contribution might room[i].experts[j].figure as
                                   room[i].experts[j].role with architects[i] choose to accompany
                                   room[i].event themed as room[i].theme and run by room[i].appointee
                                   in rooms[i]? Provide a numbered list.", temp)

7     For each i in number of rooms:
          ExportData(temp, rooms[i], architects[i], motifs[i], room[i].theme, room[i].event,
              room[i].appointee, room[i].program, room[i].experts)
```

PROMPT FORMULA IMAGE

```
Photo of [event]::[weight] in [space]::[weight] by [architect] and [appointee]::[weight]
[1 expert's name + object]::[weight] [2 expert's name + object]::[weight]
[3 expert's name + object]::[weight] ... [atmosphere]::[weight] [time of day]::[weight]
```

▲ **Fig. 35:** Overview of figures, locations, motifs, pseudo-code of the prompt cascade, formula for Midjourney V5.

ferent nature of the two computing paradigms involved; GPT 3.5 turbo uses natural language as input and responds in natural language, working probabilistically. The Wolfram Language administering prompts and responses is precise and symbolic, following a rule-based structure. Through multiple rounds of iterative testing and tweaking of our prompting templates – *prompt engineering* – we made the LLM output concise responses as numbered lists without adding lengthy introductions, explanations, or caveats. Clear instructions were given where the output needed to follow a specific format. Our code, on the other hand, was written to handle error messages, empty answers, or responses not following the right format. We iterated through all rows and columns of the table at different temperatures, saving results as tabular and text files. The final table contains *curated* suggestions across a temperature spectrum from 0,3 to 1,4; higher values would produce incoherent, cryptic or rambling output, which would also often break the prompting cascade. The result is 32 events (Nickl and Bokhari 2023) – ideated through role-played characters and having undergone only minimal 'pruning' (Allado-McDowell 2020, 93). Complex programs, spatial qualities and atmospheres emerge between the rows; the table begins to work as a screen, rendering each cell into a myriad of colours.

Imaging Impressions

Fig. 34 P. 92
Fig. 36 P. 93
Fig. 37 P. 93
Fig. 38 P. 94
Fig. 39 P. 95

To further concretise the textual outputs, we chose Midjourney V5, a model generating images from natural language text prompts (Midjourney Inc. 2023). The full detailed descriptions were too extensive for Midjourney's natural language input capabilities to satisfyingly prioritise the most important information and picture the spaces. Therefore, we prompted the LLM to enact yet another role: as a *prompt engineer* for image generation, GPT-4 was tasked with synthesising 32 texts, distilling each room into concise and targeted prompts following a given formula [Fig. 35]. This step forces the LLM to identify the most relevant information for a visual representation of space, while also assigning weights that can be interpreted by Midjourney's interface. Even when the LLM chose contributions that were not visual in nature, such as the ones by musical experts, they nonetheless contributed to the atmosphere, synaesthetically rendering each space [Fig. 37].

Casting Reflections: the Palace Talks

There are fireworks around Lantona Fountain as Icarus takes flight in an acrobatic performance around it [Fig. 38, No. 8], and Claude Nicolas Ledoux welcomes contestants to a weightlifting competition [Figs. 37 and 38, No. 1]. Buckminster Fuller prepares the Salon of Apollo as a solar eclipse observatory [Fig. 38, No. 23]. Julia Morgan addresses cosmic motifs in the Bosquets, hosting a guided meditation session between beds of wild bergamot and trees decorated with wildflower garlands [Figs. 39 and 38, No. 2]. Gian Lorenzo Bernini reenacts the eruption of Mount Vesuvius in the Salon of Diana while smoky mezcal margaritas are served in ceramic replicas of Pompeii artefacts [Figs. 34 and 38, No. 24]. And, very fittingly to the whole occasion, Paul Revere Williams invites Ray Kurzweil to chair a conference on AI in the Peace Salon, alongside

AI-generated art by Rafik Anadol and music by Holly Herndon [Fig. 38, No. 17]. It seems that *"the machines know everything, but everything to them has no meaning. And this is the part that falls to us architects: to configure meaningless global elements meaningfully in a concrete location"* (Hovestadt 2020, 561). What can we, as architects and designers, take away from Automata at Court?

Fig. 38 P. 94

Harnessing LLMs Beyond the Generic

Whereas a long, single prompt to an LLM results in fairly predictable and generic results, which remain high level, our table-based schema and programmed prompt cascades allowed the LLM to develop more detailed and specific solutions that are infinitely replicable for different spaces. Role-playing allowed us to individuate LLMs as singular machines into separate automata, which further allowed us to situate them in a larger team of intelligences collaborating towards a shared goal. Setting the professions made the LLM consider different domains important for envisioning spaces, while letting the LLM associate certain individuals with these professions made the outcomes less generic and more characterful. The outcomes, harnessing vast amounts of general knowledge stored in the LLM, are surprisingly diverse and would have been hard to ideate at the same depth, especially in the same time – computing all 32 rows at a specific temperature took about one hour.

The approach also comes with clear limitations. The pre-set schema only accommodates the preselected roles. Certain suggestions can be redundant or contradictory; and while all expert roles are aware of the program of the event, the prompt cascade collects their contributions in parallel, without the possibility for horizontal coordination. In certain cases, we faced the issue that the LLM would overly associate certain professions with the oeuvre of certain individuals, possibly due to limitations in the training data. For example, the LLM proposed Piet Oudoulf as a landscape designer for many different rows, seemingly not considering other given parameters. This observation highlights not only the importance of the body of knowledge they are trained on, but also how human individuals and their work contained in it enable the future workings of LLMs. Whether explicitly stated, as we chose to do, or implicitly internalised in their training process – as machines, they contain *"human reality, human gesture fixed and crystallized into working structures"* (Simondon 2017, 18).

Sculpting Prostheses: LLM-Aided Architectural Practice

By discretising the LLM into discrete roles, the experiment opens up new avenues for further research on collaboration across intelligences – whether residing in humans or machines – towards a cohabitation with alien intelligences, 'across the killing divisions of nature, culture, and technology and of organism, language, and machine' (Haraway 2016, 118). We conducted a small workshop; here, participants not only annotated and reviewed machine-generated suggestions on a large-scale printout but also played roles not present in the LLM *team* [Fig. 33].

Fig. 33 P. 96–97

Whether interior designers or perfumers, their contributions could either complement the general outline of a space or add an element of much-needed alienation, out of the realm of the probable. Accommodating images and text, practices of annotating or highlighting tables and reports – whether analogue or digital – could be easily adapted for a future practice in which LLMs are embedded. This includes the rethinking and redesigning of forms of collaboration between different intelligences, and *continuous negotiation* of roles and responsibilities within an AI-aided design team. On one hand, the architect becomes 'the permanent coordinator and inventor of the machines that surround him', on the other, the architect's *"presence to machines is a perpetuated invention"* (Simondon 2017, 18).

Architectonics as a General Intelligence

Automata at Court is an attempt to offer a starting point to make a new kind of collaboration with LLMs in design and architecture possible; including developing a logistic framework across domains, casting characteristic automata, and staging an architectonic stage play. While we, as architects, did not design a single contribution or artefact in this process, abstracting away from common usage paradigms allowed us to play a different role: inventing a schema, identifying decisions and making selections from the sea of characteristic outputs. This logistic framework does not suggest replacing the architect or designer but rather augmenting their capabilities by establishing conceptual communication with other characters; be they artificial, natural or even human – here, an *intelligent team* is at work. A vivid conversation takes place, between the *role-playing designer* and the *designer as role-play*. The table hosts and records the theatre of a project. Domains, objects, technologies and crafts; the digital cogs and gears, coils and weights in the architectonic model.

Endnote:

1. Both Adil Bokhari and Agostino Nickl are equal contributors to this work and designated as co-first authors.

2. This reference to 'R.U.R.' alludes to Karel Čapek's 1920 play "Rossumovi Univerzální Roboti" or "Rossum's Universal Robots," which is credited with introducing the word "robot." The robots in the play, initially created to serve humans, ultimately rebel, reflecting deep-seated anxieties about automation and artificial intelligence.